

REMARKS/ARGUMENTS

The Examiner rejected claims 11, 17, and 29 as being indefinite (35 U.S.C. §112, par. 2) on the grounds that the claim element “the cached keys” lacked antecedent basis. Applicants amended these claims to recite “cached keys”, so that the first recitation of this element provides the antecedent basis. Applicants request the Examiner to withdraw this rejection in view of these amendments.

The Examiner rejected claims 1-30 as obvious (35 U.S.C. §103(a)) over Choy (U.S. Patent No. 5,551,027), Cornwell (U.S. Pub. No. 2002/0032678), and SGI (Linux Man page for fetch). Applicants traverse these rejections for the following reasons.

Claims 1, 13, and 19 concern accessing data in a database table, and require: receiving a fetch request to fetch data from a base table that satisfies a query predicate, wherein rows of the base table are stored in table partitions and wherein there is one index partition for each determined table partition, wherein each index partition includes nodes, wherein each node in each index partition includes at least one key column value from a corresponding table row in the table partition associated with the index partition and a location identifier identifying the corresponding table row in the corresponding table partition; determining a set of nodes, one from each index partition, whose key column value satisfies the query predicate; selecting one node from the set; and returning data from the table row identified by the location identifier in the selected node in response to the fetch request.

Applicants amended claim 13 to clarify that certain limitations are in “means” form.

Choy mentions that the Global Index Table has a list of entries, each having an Index Key Value and a set of PIDs, so there is only a single entry for a given partition in which a key exists. (Choy, col. 7, lines 34-37) There is a Local Index Table for each partition, where the Global Index Table has one unique Global Index entry for each distinct Local index Key Value in each Local Index Table. Choy mentions that the Global Index is used to route an access request to the target partitions for processing. (col. 11, lines 6-20) The query is sent to each partition identified in the Global Index as having the Key Value evaluation using the Local Indexes. (col. 11, lines 37-40).

Although the cited Choy discusses the use of Local Indexes to index partitions, the Examiner recognized that the cited Choy does not teach or suggest the use of its Local Index scheme for processing fetch requests as claimed. The Examiner then cited Cornwell and SGI as

teaching the processing of fetch requests and effectively found that it would be obvious to modify Choy's use of local indexes to search partitions to apply to fetch requests. Applicants traverse.

First off, there is no suggestion in the cited art of using index partitions, one for each table partition, to search the table partitions for a fetch request query to fetch data from a base table. The Examiner cited col. 2, lines 55-59 of Choy as providing the motivation to modify the index partitioning scheme of Choy to be applied to processing fetch requests. (Office Action, pg. 4) Applicants traverse.

The cited col. 2 mentions that indexes are maintained on a search field to provide search efficiency. Applicants submit that although indexes provide search efficiency, there is no teaching or suggestion here that the particular described index scheme of Choy be used for search efficiency for fetch requests. For instance, there is no suggestion or motivation to use with search requests the claimed indexing scheme of one index partition for each determined table partition, wherein each index partition includes nodes, wherein each node in each index partition includes at least one key column value from a corresponding table row in the table partition associated with the index partition and a location identifier identifying the corresponding table row in the corresponding table partition.

Accordingly, claims 1, 13, and 19 are patentable over the cited art because the cited art does not teach or suggest the combination of claim requirements.

Claims 2-12, 14-18, and 20-30 are patentable over the cited art because they depend from claims 1, 13, and 19, which are patentable over the cited art for the reasons discussed above. Moreover, the below discussed dependent claims provide additional ground of patentability over the cited art.

Claims 2, 14, and 20 depend from claims 1, 13, and 19 and further require: determining whether to modify a direction of the fetch request, wherein the direction comprises the direction in which the index partitions are scanned to determine nodes whose key column values satisfy the query predicate; modifying the direction of the fetch request if the determination is made to modify the fetch request; and determining the set of nodes based on the direction of the fetch request.

Applicants amended these claims to further require that the direction comprises the direction in which the index partitions are scanned to determine nodes whose key column values

satisfy the query predicate. This additional requirement is disclosed on at least para. [0027], pg. 11.

The Examiner cited col. 8, lines 31-35 of Choy as teaching the claim requirement of determining whether to modify a direction of a fetch request. (Office Action, pg. 4) Applicants traverse. The cited col. 8 mentions that the global index is used to associate the individual index key values with their target partitions, so that an access request with a partition selective search predicate on the index key can be directed to the target partitions for processing.

Nowhere does this cited col. 8 anywhere teach or suggestion modifying a direction of the fetch request, which is the direction in which the index partitions are scanned. Instead, the cited col. 8 discusses how the global index is used to direct an access request to the appropriate index having an index key subject to the access request. There is no suggestion of the claim requirement of determining whether to modify the direction in which the index partitions are scanned.

The Examiner cited col. 9, lines 11-16 of Choy as teaching modifying the direction of the fetch request, which is the direction in which the index partitions are scanned. (Office Action, pg. 4) Applicants traverse.

The cited col. 9 mentions that the global index may contain the identifiers of indexed records to allow skipping local index table access. There is no teaching or suggestion in the cited col. 9 of modifying the direction of the fetch request, which is the direction in which the index partitions are scanned.

The Examiner found that Choy does not teach modifying the direction of a fetch and proposed modifying Choy with the fetch requests of SGI . However, Applicants submit that the cited Choy does not teach modifying the direction in which an index is scanned for any type of request, not just a fetch request. Thus, even if one were to modify the cited Choy to be applied to fetch requests as the Examiner proposes (which is improper for the reasons discussed above), such proposed modification still does not teach the claim requirement of determining and modifying the direction of a fetch request, which is the direction in which the index partitions are scanned.

Accordingly, amended claims 2, 14, and 20 provide additional grounds of patentability over the cited art because the additional requirements of these claims are not taught or suggested in the cited art.

Claims 3 and 21 depend from claims 2 and 20 and further recite determining whether to modify the direction of the fetch request is based on a current fetch direction and whether the current fetch direction is opposite an ordering of the index partitions.

The Examiner cited para. [0094] of Cornwell as teaching the additional requirements of these claims. (Office Action, pg. 5). Applicants traverse.

The cited para. [0094] discusses fetching backward in a base table from a current position on the page by positioning the cursor back from the current position by a number of rows. If the cursor cannot be positioned backwards by the requested number of rows, then the cursor goes to the page including the previous sequential rows in the base table.

The cited para. [0094] discusses how one fetches backwards in a result table. Nowhere does the cited para. [0094] anywhere teach or suggest the claim requirement of determining whether to modify the direction of a fetch request, wherein the direction indicates the direction in which the index partition is scanned. Further, nowhere does the cited para. [0094] anywhere teach, suggest or mention making such determination based on whether a current fetch direction is opposite an ordering of the index partitions. Instead, the cited para. [0094] discusses the direction in which rows of a result table are accessed, not changing the direction in which index partitions are scanned to determine nodes whose key column values satisfy the query predicate.

Accordingly, claims 3 and 21 provide additional grounds of patentability over the cited art because the additional requirements of these claims are not taught or suggested in the cited art.

Amended claims 4, 15, and 22 depend from claims 2, 14, and 20 and further require that modifying the direction of the fetch request comprises: setting the fetch direction to backward if the fetch direction is backward and the fetch direction is not opposite the ordering of the index partitions or if the fetch direction is forward and the fetch direction is opposite the ordering of the index partitions; and setting the fetch direction to forward if the fetch direction is backward and the fetch direction is opposite the ordering of the index partitions or if the fetch direction is forward and the fetch direction is not opposite the ordering of the index partitions.

Applicants amended these claims to clarify that the ordering is of the index partitions.

The Examiner cited para. [0099] of Cornwell as teaching the additional requirements of these claims. (Office Action, pg. 5) Applicants traverse for the following reasons.

The cited para. [0099] discusses a fetch absolute operation, which is the number of rows to fetch forward or backward from the first entry in the result table. This is performed by determining the distance from the current entry to the requested entry, and then convert this command to a fetch relative to fetch to the requested position.

Nowhere does the cited para. [0099] anywhere teach or suggest modifying the direction of the fetch request based on the partition index ordering. Instead, the cited para. [0099] discusses how to fetch to an absolute requested row from the current position, not to change the direction of a fetch based on the index ordering. Further, there is no teaching or suggestion in the cited para. [0099] of considering an index ordering when determining how to access a row at an absolute position in a result table.

Accordingly, claims 4, 15, and 22 provide additional grounds of patentability over the cited art because the additional requirements of these claims are not taught or suggested in the cited art.

Claims 5 and 23 depend from claims 2 and 20 and further require that if the fetch request is a first fetch of the fetch request, then selecting one node starting from one of: a lowest key value from each index partition if the fetch direction is forward or highest key value from each index partition if the fetch direction is backward.

The Examiner cited para. [0095] as teaching the additional requirements of these claims. (Office Action, pgs. 5-6) Applicants traverse.

The cited para. [0095] discusses how to go to a previous page to fetch backwards sequential rows in the result table if the requested row is not on the current page. Nowhere does the cited para. [0095] anywhere teach or suggest selecting a node from a lowest or highest key value from each index partition depending on the fetch direction. There is no mention in the cited para. [0095] of selecting key values from index partitions as claimed. Instead, the cited [0095] discusses how to go to a page having the previous sequential rows in a result table to fetch backward.

Accordingly, claims 5 and 23 provide additional grounds of patentability over the cited art because the additional requirements of these claims are not taught or suggested in the cited art.

Claims 6-10, 16, and 24-29 provide further limitations concerning the direction of the fetch request and index partitions. The Examiner cited other sections of Cromwell as teaching

the additional requirements of these claims that concern how to fetch in different directions in a result table to retrieve a row from a result table. Nowhere does the cited Cromwell teach or suggest the additional requirements of these claims providing additional requirements concerning index partitions on table partitions. Instead, the cited Cromwell discusses how to fetch forward or backward through a result table whose sequential rows are on multiple pages.

Amended claims 11, 17, and 29 depend from claims 1, 13, and 19 and further require discarding the cached keys if the fetch request is in an opposite direction of a previous fetch request; determining a new set of nodes from each index partition; and caching the determined new set of nodes when performing the fetch operation.

The Examiner cited para. [0095] of Cromwell as teaching discarding the cached keys if the fetch request is in an opposite direction of a previous fetch request. (Office Action, pg. 8) Applicants traverse.

The cited para. [0095] discusses how to go to a previous page to fetch backwards sequential rows in the result table if the requested row is not on the current page. Nowhere does the cited para. [0095] anywhere teach or suggest discarding cached keys if the fetch request is in an opposite direction of a previous fetch request. The cited Cromwell discusses how to scroll to a page when fetching backward. However, there is no mention or teaching of the claim requirement of discarding cached keys if the fetch request is in an opposite direction of the previous fetch request. Moreover, there is no teaching in the cited Cromwell of considering the direction of the previous fetch request when processing a current fetch backward request.

Accordingly, claims 11, 17, and 29 provide additional grounds of patentability over the cited art because the additional requirements of these claims are not taught or suggested in the cited art.

Applicants amended claims 12, 18, and 30 to depend from claims 11, 17, and 29.

Conclusion

For all the above reasons, Applicant submits that the pending claims 1-30 are patentable over the art of record. Applicants have not added any claims. Nonetheless, should any additional fees be required, please charge Deposit Account No. 09-0460.

The attorney of record invites the Examiner to contact him at (310) 553-7977 if the Examiner believes such contact would advance the prosecution of the case.

Dated: May 15, 2006

By: /David Victor/

David W. Victor
Registration No. 39,867

Please direct all correspondences to:

David Victor
Konrad Raynes & Victor, LLP
315 South Beverly Drive, Ste. 210
Beverly Hills, CA 90212
Tel: 310-553-7977
Fax: 310-556-7984